

Claims

[c1] 1. A method and a system for integrating and coordinating resources using the internet or intranets for synchronous and non synchronous communications by displaying Content;

wherein said resources are any devices, applications, programs, ActiveX, or embedded objects that could receive directly or indirectly commands to perform actions related to a Content; wherein said Content is any data or reference to data that could be used by a resource; wherein said data could be static or produced live; wherein displaying is the process of building at run time and sending for each party involved in the communication a personalized Order that when executed, Commands are sent directly or indirectly to the resources available for the respective party that will closest match a desired action; wherein a Command execution could entail setting properties, calling methods or functions, or triggering events in a resource; wherein instantiation of a resource with the respective parameters or arguments is considered also sending a Command; wherein coordinating refers to the fact that more

than one of said resources could be involved in the displaying of said Contents, and that said resources are not necessarily the same in all the parties involved in the communication and is expressed in the complexity and uniqueness of each of said Orders;

[c2] 2. A method and a system based on claim 1 that allows using just four basic elements in the User Interface: Contents, Sessions, Users and Groups, and different kinds of relationships among those four elements to configure and manage the whole system;

wherein mentioned parties involved in the communication are either Users or Sessions; wherein mentioned data is provided by Servers; wherein Session-Id is a key used at the server side to group all the resources connected directly or indirectly to any Server from the same computer; it is a function of any server integrated to the system to identify the Session in a connection; wherein Sessions are opened every time someone connects for the first time and closed for good after a timeout without any connection to any server integrated to the system; wherein User is a known individual or a Session that has established a relationship to a User; when a Session logs in, the Session belongs to the User; wherein virtual User accounts could be created for devices that

do not have a relationship to real People; wherein opened Sessions are used for synchronous communications; wherein Users are used for non-synchronous communications; wherein Groups are groups of either Users or Sessions; wherein Contents in relationship to Sessions, Users and Groups modify their respective properties; wherein Contents in relationship to other Contents increase their complexity; wherein mentioned personalized Orders are documents produced according to the Sessions, Users, Groups, and Contents directly or indirectly involved in the communication and the system configuration; wherein said system configuration refers to a Table with information about what resources can handle what Display-Options, and a Table of fallback that indicates in case there is no resource that can handle selected Display-Option, what Display-Option should be used next because probably is also next in matching the intended action; wherein said Display-Option refers to a particular way the handling of the data should be done; wherein said system configuration does not need to be changed, and therefore is not needed in said User Interface of the system. wherein each of said Servers updates the Session to Content relationship data, so the history of a Session in relationship to each Content, including responses

submitted for Question-Contents can be tracked;

- [c3] 3. A method and a system based on claim 1, where a Browser becomes a bi-directional communication resource independent from the mechanism used to pull or push commands to said Browser;
- wherein commands are embedded in the HTML responses for Browser's requests; wherein the Commands are according to said Browser's capabilities like javascript and java enabled; wherein several options can be used to force said Browser's request without a User's input, some of said options are: (a) The use of HTML-Tag like <META HTTP-EQUIV="Refresh"> to pull said commands; (b) The use of an Applet that maintains a connection to the system all the time pushing commands in the following way: when the Display Order is built for a Session that indicates that there is an Applet connection, it includes said command to said Browser, and an additional command to said Applet to execute showDocument() and force said Browser to request a page and therefore read the command; (c) The use of an application that maintains a connection to the system pushing commands to said Browser in the same way as in said Applet but by executing a DDE pop-up instead of a showDocument(); wherein all commands

sent to the Browser could be executed by just one retrieval of an HTML-page from said Browser; wherein said Commands to the Browser have a Name, Body and Header and when said Commands are merged in the HTML-response only one Body, and all the Headers are added for all Commands with the same Name.

- [c4] 4. A method and a system to transform Orders at run-time where they could get a life of their own; wherein said Orders are XML documents that are executed by an XML-Processor; wherein said Orders are made out of nodes called Item; wherein said Items have integer, string and XPath (reference to nodes) Variables, and they could be global to the whole Order, local to the Item, and from a Response document passing through the system; wherein said Items have a unique name attribute; wherein said Items have OnExecute, OnResponse, and OnTimeout nodes that are executable; wherein the execution of said executable Nodes deals with the Update, CallItems, and Evaluate/Otherwise Nodes; wherein CallItems carry a list of the names of the Items that should be executed; wherein Evaluate/Otherwise allows the comparison between Variables (greater than, less than or equal to), if true then the Childs of the Evalu-

ate, if false the Childs of the Otherwise node are executed recursively; wherein Update node carry information to perform operations of Variables, mathematical (+,-,* ,/,=), string (+,=) and node operations like replace, delete, add and copy; wherein node operations could modify the Order itself; wherein OnExecute is executed when the Order is received if its parent Item is of type start, or if its parent Item's name is in the CallItems list; wherein at execution the Item is transformed with the XSL indicated in the SendTransform into a Command or another Order that is sent on the channel indicated on SendTopic nodes; wherein the XML-Processor waits for a response only if there is a ResponseTimeout and a Responseld for the executed Item; wherein an Item waiting for a response becomes a Task; wherein On-Response is executed when a document passing through the system matches the Responseld's XPath list of the Task; wherein OnTimeout is executed, if there is not a response after time indicated in ResponseTimeout is elapsed.

- [c5] 5. A method and a system based on claim 4, to handle Events using Content; wherein said Events are any documents, either Orders or Commands processed by the XML-Processor;

comprising the steps:(a) enrolling the EventListeners by using the XML-Processor interface and passing an XML-header, an XPath list, a reference to an XSL-Content, and a reference id; (b) transforming said document together with said XML-header with said XSL-Content into an Order, when all said XPath in said list return valid when applied to a document passing through the system; (c) said Order including Commands to the required resources and Server side components to handle the Event; wherein said method and system allows to have in general any Application monitoring the system comprising the steps of: (a) creating all XSL-Contents needed to transform the Events to be monitored; (b) said application creating a channel to communicate and creating a thread to process the messages received in said channel; (c) the application enrolling the EventListeners with the respective XSL-Content reference, XML-Header, XPath list and reference id; (d) the application handling the commands received in said opened channel product of the Orders produced when said Events are detected; wherein said methods and systems allow in particular to have an OnEvent handler for HTML pages, which means that with a Browser that is JavaScript and frames enabled, an HTML page could be updated according to Events without the

need of reloading the whole page; Said method comprising the steps of: (a) creating all XSL-Contents needed to transform the Events to be monitored; (b) every time said page is requested, enrolling all the required EventListeners with the respective XSL-Content reference, XML-header, XPath list and reference id;(c) said XSL-Content producing Orders with commands for the Browser that include JavaScript functions that when executed in the frame that receives said commands change said page loaded in another frame;(d) every time said page is unloaded, removing the EventListeners with the respective reference id.

[c6] 6. A method and a system based on claim 1 and 4 for caching Content;

wherein with a Client Application following the steps of: (a) said Client Application connects to a server integrated to the system and identifies the user it belongs to; (b) said Application using said connection to check periodically in the list of Contents posted for said User and in the list of Contents to be synchronized in the Groups where said User belongs to for Contents to be downloaded; (c) said Client Application starting the downloading of said Contents if not found or out of synch in the local machine; (d)

mentioned XML-Builder producing a display Order for said User that when said User's Client Application is connected and the Content is cacheable will indicate to the resource to retrieve the data from said User's local Client Application and not a remote server; (e) wherein said Client Application being able of retrieving the Content's data from the server at the same time the data is sent to the resource; (f) wherein the Client Application could enroll EventListeners mentioned in Claim 5, so that it starts caching Contents at the moment they are posted and be ready for when they are requested.

[c7] 7. A method and system based on claim 1 and 4 to automate handling of Sessions connected to the system according to the display of Contents in specific rolls in the different relationships to Groups, Sessions, Users and other Contents;

wherein Events could trigger the displaying of Content in said rolls; wherein by using Conditional-Content in said rolls it is possible to personalize and add intelligence to the interaction between the system and the session; wherein said Conditional-Content carry information regarding an SQL statement, the database to apply said statement and an XSL-Content reference to transform the SQL result;

wherein said SQL result, an XML document, is transformed with said XSL producing a document that contains a list of Contents; wherein said list of Contents replaces said Conditional-Content with relevant information at run-time; wherein the display of an XSL- Content could entail the creation of Orders; wherein the interaction could refer, but is not limited to: (a) transferring Sessions; (b) notifying Users that someone wants to communicate; (c) notifying Sessions that they are on hold, have been accepted to a Group, please leave a message, sorry I am busy, please call later and many others; (d) sending questionnaires; (e) displaying other contents according to responses to questionnaires; (f) displaying content according to information stored in the database; (g) processing payments.

[c8] 8. A method of organizing objects in three distinct sections corresponding to the relationships between a list of Sessions and a list of Contents or a list of Users and said list of Contents (refer to *FIG. 7*);

wherein said sections are Header/Column, Row/Reference, and Row/Column; wherein said objects according to their embodiment and alignment provide information and can be used to trigger actions; wherein said Header/Column section further com-

prising of a plurality of Header/Column-n sub-sections; wherein the embodiment of the objects in said sub-section provide information related to Content-n and can be used to trigger actions related to Content-n, Content-n involving all the Sessions and Content-n involving all the Users related to said Sessions; wherein said Row/Reference section further comprises a plurality of Row-m/Reference sub-sections; wherein the embodiment of objects in said sub-section provide information related to Session-m and can be used to trigger actions related to Session-m and if Session-m is logged in provide information related to the respective User and trigger actions regarding said User; wherein said Row/Column section further comprising of a plurality of Row-m/Content-n sub-sections; wherein the embodiment of the objects in said sub-section provide information related to the relationship of Session-m with Content-n and can be used to trigger actions related to that specific relationship and if Session is logged in provide information related to the relationship of User-m with Content-n and can be used to trigger actions related to their respective relationship; wherein said Row-m/Content-n sub-section is aligned to the respective Row-m/Reference and Header/Content-n sub-sections. wherein in the case

of Users instead of Sessions Objectives could be placed in the three distinct sections in relationship to the Users; wherein said sections are Header/Column, Row/Reference, and Row/Column; wherein said objects according to their embodiment and alignment provide information and can be used to trigger actions; wherein said Header/Column section further comprising of a plurality of Header/Column-n sub-sections wherein the embodiment of the objects in said sub-section provide information related to Content-n and can be used to trigger actions related to Content-n, Content-n involving all the Users and Content-n involving all the Sessions that are open for respective Users; wherein said Row/Reference section further comprising of a plurality of Row-m/Reference sub-sections; wherein the embodiments of objects in said sub-section provide information related to User-m and a plurality of Sessions related to User-m and can be used to trigger actions related to said User-m and said Sessions; wherein said Row/Column section further comprising of a plurality of Row-m/Content-n sub-sections; wherein the embodiment of the objects in said sub-section provide information related to the relationships of User-m with Content-n and the relationships of a plurality of Sessions related to User-m with Content-n; wherein said ob-

jects in the Row-m/Content-n sub-section can be also used to trigger actions related to their respective relationship; wherein said Row-m/Content-n sub-sections are aligned to the respective Row-m/Reference and Header/Content-n sub-sections.

- [c9] 9. A method and a system based on claim 8 to identify and get information about Sessions, Users connected at the moment and/or in the past;
wherein the method of organizing Objects could be applied to an active or closed Session, or a User and any to all the Contents it has had a relationship with; wherein said plurality of Row/Content-n sub-sections link to information related to the relationships of said Session and/or User with each Content; wherein said plurality of Row/Reference sub-sections link to all sorts of information related to said Session and/or User;
- [c10] 10. A method and a system to calculate final Grades using Content based on the value-weight that can be entered for each Content-Child of a Content;
wherein the grade for a specific Question-Content goes from 0 to 100, and can be graded manually or automatically; wherein the final grade is obtained with the result of a recursive evaluate function that has as parameters the maximum score, the Content

and the Session for whom the final grade is calculated; said function first adds all the value-weights of the Childs of the first generation (direct Childs) in to a total-value-weight, then for each of said Child Contents checks if it has more Childs; if said Child Content does have more Childs the result of a recursive call to the evaluate function is added to the final grade; said recursive call gets the respective value-weight divided by said total-value-weight and multiply by maximum score as the new maximum score, said Content Child and said Session; if said Content Child does not have more Childs the value-weight divided total-value-weight and multiplied by the score for respective Content and Session is added to the final grade. wherein instead of Session there are Users; wherein the grade is calculated base on Sessions related to said Users and a selection criteria for said Sessions in relationship to respective Content based on but not limited to: the Session with the last relationship to the respective Content; the Session with the relationship to respective Content with the best score; the Session with a relationship the respective Content where the response was entered in a specific date range; the Session chosen based on a combination of previous criteria; an average of all Sessions with a relationship to respective Content.

[c11] 11. A method and a system to set rules for Users to post any type of messages using Content;
wherein a User can receive messages according to the following address structures: x.u.p@server; x.u@server; or x@server; wherein x, u, p are strings of any length of any characters or numbers not including special characters defined in the configuration file; wherein x refers to the User-account field for the record of the User receiving the message; wherein u refers to a unique Content in a specific roll for the User sending the message; and wherein p refers to an additional Content in the same roll for the User sending the message; wherein the identity of the receiver is defined by the unique identifier x, and the identity of the sender is defined by the unique "u" or "u" and "p" strings; wherein in the case of emails and SMTP servers the address structure can be taken literally; wherein in case of other servers the address structure may not be taken literally: in the case of a voice mail, for example, "x" refers to a phone number/extension, "u" and "p" are gathered through DTMF, by pressing additional keys after respective prompts and the server is the voicemail system itself; wherein the server receiving the message gathers additional information from the connection,

or the message itself; wherein the identity of the sender could be also established with mentioned additional information, like the senders email address or caller id; wherein a message could be classified in three different ways: as coming from a know User, a not known User or an Invalid identification; wherein Content-Mailboxes are set to receive messages according to the identity of the sender and receiver, the mentioned three classifications, and the mentioned additional information gathered by the serves; said Content-Mailboxes hold information on how to handle the message particularly in regards to accepting, or rejecting the message and penalizing the connection by delaying the final termination; wherein posting messages to said Content-Mailboxes could trigger Events that could be used to notify target User live, and in the case of bidirectional communications he/she could display specific Content to "answer"/establish a synchronous communication; wherein Content-Mailbox in specific rolls for a User establish the permissions to see/retrieve or not the message; wherein placing a message in a Content-Mailbox that is not in the roll Content to User that indicates it is accessible to be read, is a way of filtering out messages for a User.

[c12] 12. A method and a system to manage appointments using the Internet based on Content, Users and Groups; wherein the person asking for an appointment and the person giving an appointment have User accounts created in the system; wherein Users providing appointments build schedules for different types of appointments; wherein said schedules for appointments are made based on Users giving appointments, the type of appointment, date range, time range; wherein the view of said schedule is built using said XML-Content, XSL-Content to transform the output and the User requesting the appointment and related Contents; wherein different types of servers could use same XML-Content and respective XSL-Content to display same schedule in their respective formats (HTML or WML for example); wherein a relationship between a Group(s) and an XML-Content that represents said schedules gives the Users belonging to mentioned Group(s) access to said schedules;